Tablas de dispersión (Tablas de Hash)

Tablas de dispersión

- La dispersión es una técnica empleada para realizar inserciones, eliminaciones y búsquedas en un tiempo promedio constante.

- Un uso común de esta técnica lo representan los diccionarios. Un diccionario almacena objetos formados por una clave, por la cual se busca en el diccionario, y su definición, que es lo que se devuelve.

Tablas de dispersión

- La estructura de datos ideal para la tabla de dispersión es simplemente una lista de tamaño fijo N que contiene las claves.
  - Cada clave se hace corresponder con algún número en el intervalo entre 0 y N −1, y se coloca en la celda correcta.
  - A la correspondencia se le denomina función de dispersión.

Funciones de dispersión

- Toda función de dispersión debe:
  - calcularse de forma sencilla O(1).
  - distribuir uniformemente las claves.

- Por ejemplo, si las claves son números enteros, (clave % N) es una función buena, salvo que haya propiedades indeseables:
  - Si N fuese 100 y todas las claves terminasen en cero, esta función de dispersión sería una mala opción.
    - Es buena idea asegurarse de que el tamaño de la tabla sea un número primo.

Resolución de colisiones: dispersión abierta

- La estrategia consiste en tener una lista de todos los elementos que se dispersan en el mismo valor.
- Para efectuar una búsqueda, usamos la función de dispersión para determinar qué lista recorrer.
- Para efectuar un insertar, recorremos la lista adecuada para revisar si el elemento ya está en la lista. Si el elemento resulta ser nuevo, se inserta al frente o al final de la lista.
- Además de las listas enlazadas, se podría usar un árbol binario de búsqueda o AVL.

Ejemplo de colisiones: dispersión abierta

Valores de la función de dispersión:

hash(Ana,11) =7	hash(Luis,11)=6	hash(José,11)=7
hash(Olga,11)=7	hash(Rosa,11)=6	hash(Iván,11)=6

Tabla después de insertar *Ana*:

0	1	2	3	4	5	6	7	8	9	10
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[Ana]	[ ]	[ ]	[ ]

Tabla después de insertar *Luis*:

0	1	2	3	4	5	6	7	8	9	10
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[Luis]	[Ana]	[ ]	[ ]	[ ]

Tabla después de insertar *José*:

0	1	2	3	4	5	6	7	8	9	10
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[Luis]	[Ana; José]	[ ]	[ ]	[ ]

Ejemplo de colisiones: dispersión abierta

Tabla después de insertar *Olga*:

0	1	2	3	4	5	6	7	8	9	10
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[Luis]	[Ana; José; Olga]	[ ]	[ ]	[ ]

Tabla después de insertar *Rosa*:

0	1	2	3	4	5	6	7	8	9	10
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[Luis; Rosa]	[Ana; José; Olga]	[ ]	[ ]	[ ]

Tabla después de insertar *Iván*:

0	1	2	3	4	5	6	7	8	9	10
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[Luis; Rosa; Iván]	[Ana; José; Olga]	[ ]	[ ]	[ ]

Resolución de colisiones: dispersión cerrada

- En un sistema de dispersión cerrada, si ocurre una colisión, se intenta buscar celdas alternativas hasta encontrar una vacía.
  - Se busca en sucesión en las celdas $d_0(x), d_1(x), d_2(x), ...$ donde:

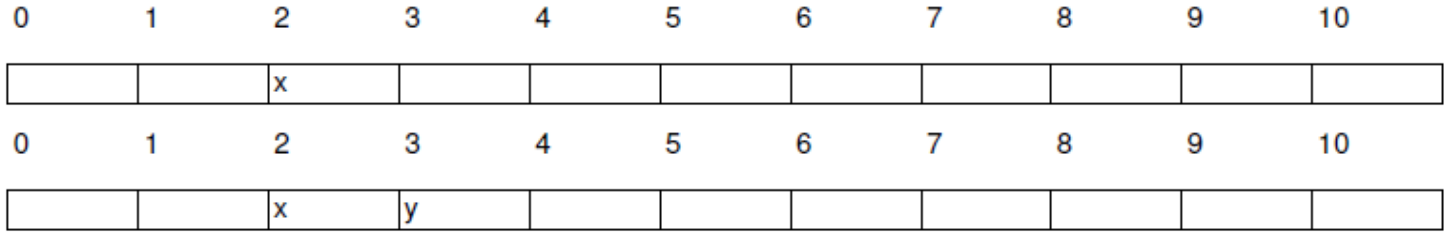
$$d_i(x) = (dispersion(x) + f(i)) \ \% \ n, con \ f(0) = 0$$

  - La función f es la estrategia de resolución de las colisiones.

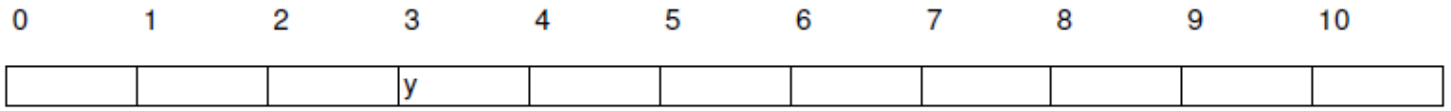
- Como todos los datos se guardan en la tabla, esta tiene que ser más grande para la dispersión cerrada que para la abierta.

Resolución de colisiones: dispersión cerrada

- La eliminación estándar no es realizable con dispersión cerrada.
  - La celda ocupada pudo haber causado una colisión en el pasado. Por ejemplo, con exploración lineal, si las claves “x” e “y” se dispersan a la misma posición (p. ej. 2):
    - Insertamos ambas claves



    - Borramos la primera



    - Si buscamos “y” no la encontrariamos
- Las tablas de dispersión cerrada requieren eliminación perezosa, aunque no haya realmente “pereza”.

Dispersión cerrada con exploración lineal

- Aquí la estrategia de resolución de las colisiones es una función lineal de i, por lo general f (i) = i.
  - Esto equivale a buscar secuencialmente en la lista (en forma circular) una posición vacía.
  - Si la tabla es suficientemente grande, siempre se encontrará una celda vacía.
    - Pero ello puede tomar demasiado tiempo.

Ejemplo dispersión cerrada con exploración lineal

Valores de la función de dispersión:

hash(Ana,11) =7	hash(Luis,11)=6	hash(José,11)=7
hash(Olga,11)=7	hash(Rosa,11)=6	hash(Iván,11)=6

Tabla de dispersión después de insertar *Ana*:

0	1	2	3	4	5	6	7	8	9	10
							Ana			

Tabla de dispersión después de insertar *Luis*:

0	1	2	3	4	5	6	7	8	9	10
						Luis	Ana			

Tabla de dispersión después de insertar *José*:

0	1	2	3	4	5	6	7	8	9	10
						Luis	Ana	José		

Ejemplo dispersión cerrada con exploración lineal

Valores de la función de dispersión:

hash(Ana,11) =7
hash(Olga,11)=7

hash(Luis,11)=6
hash(Rosa,11)=6

hash(José,11)=7
hash(Iván,11)=6

Tabla de dispersión después de insertar *Olga*:

0	1	2	3	4	5	6	7	8	9	10
						Luis	Ana	José	Olga	

Se está formando un agrupamiento primario

Tabla de dispersión después de insertar *Rosa*:

0	1	2	3	4	5	6	7	8	9	10
						Luis	Ana	José	Olga	Rosa

Tabla de dispersión después de insertar *Iván*:

0	1	2	3	4	5	6	7	8	9	10
Iván						Luis	Ana	José	Olga	Rosa

Dispersión cerrada con exploración cuadrática

- La función de resolución de colisiones es cuadrática, por lo general: $f(i) = i^2$.
- Con exploración lineal es malo llenar la tabla, porque se degrada el rendimiento. Para la exploración cuadrática, la situación es más drástica:
  - Con más de la mitad de la tabla ocupada, no hay garantías de encontrar una celda vacía.

Ejemplo dispersión cerrada con exploración cuadrática

Valores de la función de dispersión:

hash(Ana,11) =7 hash(Luis,11)=6 hash(José,11)=7

hash(Olga,11)=7 hash(Rosa,11)=6 hash(Iván,11)=6

Tabla de dispersión después de insertar *Ana*:

0	1	2	3	4	5	6	7	8	9	10
							Ana			

Tabla de dispersión después de insertar *Luis*:

0	1	2	3	4	5	6	7	8	9	10
						Luis	Ana			

Tabla de dispersión después de insertar *José*:

0	1	2	3	4	5	6	7	8	9	10
						Luis	Ana	José		

Ejemplo dispersión cerrada con exploración cuadrática

Valores de la función de dispersión:

hash(Ana,11) =7
hash(Olga,11)=7

hash(Luis,11)=6
hash(Rosa,11)=6

hash(José,11)=7
hash(Iván,11)=6

Tabla de dispersión después de insertar *Olga*:

0	1	2	3	4	5	6	7	8	9	10
Olga						Luis	Ana	José		

Tabla de dispersión después de insertar *Rosa*:

0	1	2	3	4	5	6	7	8	9	10
Olga						Luis	Ana	José		Rosa

Tabla de dispersión después de insertar *Iván*:

0	1	2	3	4	5	6	7	8	9	10
Olga				Iván		Luis	Ana	José		Rosa

Dispersión cerrada con exploración doble

- Para la resolución de colisiones aplicamos una segunda función de dispersión, en general: f (i) = i * $h_2(x)$.
  - La función nunca debe evaluarse a cero.
  - Y es importante que todas las celdas puedan ser intentadas.
    - Una función como $h_2(x) = R * (x \% R)$, con R un número primo menor que el tamaño de la tabla, funcionará bien.
    - Hay que asegurarse de que el tamaño de la tabla sea primo.

Ejemplo dispersión cerrada con exploración doble

Valores de la función de dispersión:

	Ana	Luis	José	Olga	Rosa	Iván
$h_1(x, 11)$	7	6	7	7	6	6
$h_2(x, 11) = 5 - h_1(x, 11)\,\%5$	3	4	3	3	4	4

Tabla de dispersión después de insertar *Ana*:

0	1	2	3	4	5	6	7	8	9	10
							Ana			

Tabla de dispersión después de insertar *Luis*:

0	1	2	3	4	5	6	7	8	9	10
						Luis	Ana			

Tabla de dispersión después de insertar *José*:

0	1	2	3	4	5	6	7	8	9	10
						Luis	Ana			José

Ejemplo dispersión cerrada con exploración doble

Valores de la función de dispersión:

$h_1(x,11)$
$h_2(x,11) = 5 - h_1(x,11)\%5$

Ana	Luis	José	Olga	Rosa	Iván
7	6	7	7	6	6
3	4	3	3	4	4

Tabla de dispersión después de insertar *Olga*:

0	1	2	3	4	5	6	7	8	9	10
		Olga				Luis	Ana			José

Tabla de dispersión después de insertar *Rosa*:

0	1	2	3	4	5	6	7	8	9	10
		Olga	Rosa			Luis	Ana			José

Tabla de dispersión después de insertar *Iván*:

0	1	2	3	4	5	6	7	8	9	10
Iván		Olga	Rosa			Luis	Ana			José

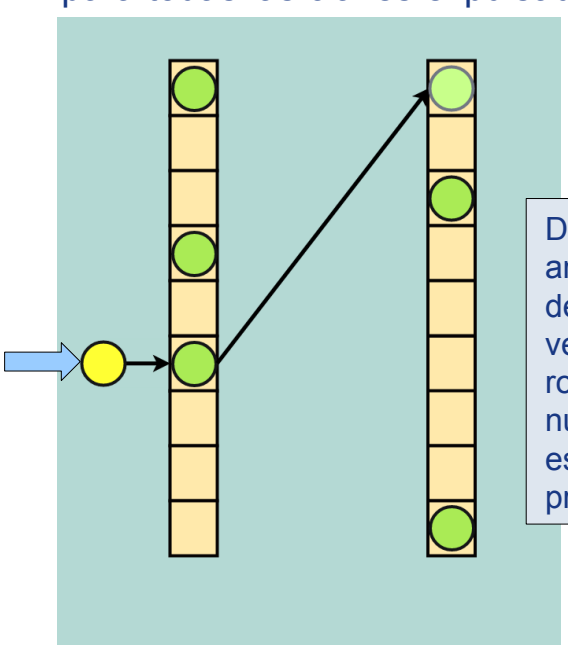
Dispersión del cuco

- El nombre deriva del comportamiento de algunas especies de cuculidae, donde la cría del cuco empuja los otros huevos o crías del nido cuando incuba.
- La inserción de una nueva clave en una tabla de dispersión puede empujar a otra clave para una ubicación diferente en la tabla.
- La idea básica es usar dos funciones de dispersión en lugar de sólo una. Esto proporciona dos posibles ubicaciones en la tabla para cada clave única y tiempos constantes en la búsqueda de valores.
- La implementación mas común es la de dividir la tabla de dispersión en dos tablas más pequeñas de igual tamaño, y cada función dispersión proporciona un índice en una de estas dos tablas.

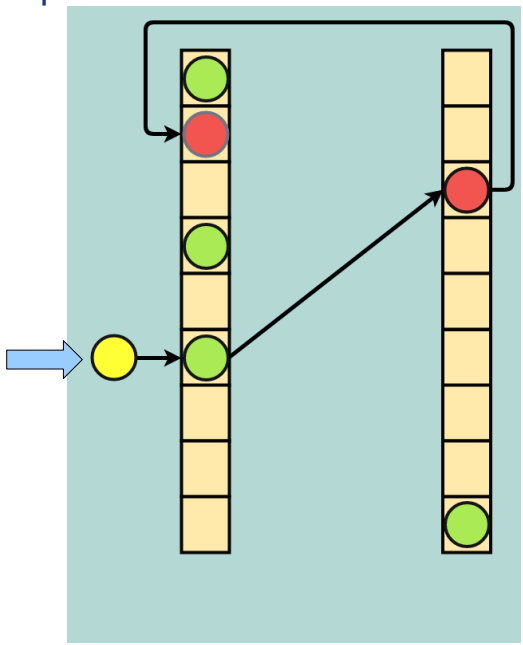
Dispersión del cuco

- Cuando se inserta una nueva clave, si esta operación no produce colisión se realiza sin inconvenientes.
- Si existen valores ocupando una de sus dos posibles ubicaciones, "se patea", es decir, desplazar, cualquier clave que podría residir en estas ubicaciones y se inserta esta clave desplazada en su lugar alternativo.
- El procedimiento de expulsión continua hasta que se encuentre un puesto disponible para todas las claves expulsadas, o se genere un loop infinito.

El dato amarillo desaloja al dato verde y este encuentra un nuevo lugar en el espacio de direcciones secundario.



Doble desalojo: El dato amarillo entrante desaloja al verde; el verde desaloja al rojo; y rojo encuentra un nuevo lugar en el espacio de dirección principal.



Dispersión del cuco

Valores de las funciones de dispersión:

h1(key) = key%11

h2(key) = (key/11)%11

	20	50	53	75	100	67	105	3	36	39
h_1(key)	9	6	9	9	1	1	6	3	3	6
h_2(key)	1	4	4	6	9	6	9	0	3	3

Se comienza insertando el 20 en la primera tabla en la posible posición indicada por h_1(20):

table[1]	-	-	-	-	-	-	-	-	-	20	-
table[2]	-	-	-	-	-	-	-	-	-	-	-

Sigue el 50:

table[1]	-	-	-	-	-	-	50	-	-	20	-
table[2]	-	-	-	-	-	-	-	-	-	-	-

Sigue el 53, pero su ubicación esta ocupada por el 20, entonces el 53 se ubica en la tabla 1 y el 20 en la tabla 2:

table[1]	-	-	-	-	-	-	50	-	-	53	-
table[2]	-	20	-	-	-	-	-	-	-	-	-

Dispersión del cuco

Valores de las funciones de dispersión:

h1(key) = key%11

h2(key) = (key/11)%11

	20	50	53	75	100	67	105	3	36	39
h_1(key)	9	6	9	9	1	1	6	3	3	6
h_2(key)	1	4	4	6	9	6	9	0	3	3

Sigue el 75, pero el 53 ocupa su lugar, Entonces el 75 se almacena en la tabla 1 y el 53 en la tabla 2.

table[1]	-	-	-	-	-	-	50	-	-	75	-
table[2]	-	20	-	-	53	-	-	-	-	-	-

Sigue el 100.

table[1]	-	100	-	-	-	-	50	-	-	75	-
table[2]	-	20	-	-	53	-	-	-	-	-	-

Sigue el 67, pero el 100 ocupa su lugar, Entonces el 67 se almacena en la tabla 1 y el 100 en la tabla 2.

table[1]	-	67	-	-	-	-	50	-	-	75	-
table[2]	-	20	-	-	53	-	-	-	-	100	-

Dispersión del cuco

Valores de las funciones de dispersión:

h1(key) = key%11

h2(key) = (key/11)%11

	20	50	53	75	100	67	105	3	36	39
h_1(key)	9	6	9	9	1	1	6	3	3	6
h_2(key)	1	4	4	6	9	6	9	0	3	3

Sigue el 105, pero el 50 ocupa su lugar, entonces el 105 se ubica en la tabla 1 y el 50 en la tabla 2. Ahora el 53 ha sido desplazado a la tabla 1 en la posición 9 y el 75 a la tabla 2 en la posición 6.

table[1]	-	67	-	-	-	-	105	-	-	53	-
table[2]	-	20	-	-	50	-	75	-	-	100	-

Sigue el 3.

table[1]	-	67	-	3	-	-	105	-	-	53	-
table[2]	-	20	-	-	50	-	75	-	-	100	-

Sigue el 36. Quedando el 36 en la posición 3 de la tabla 1 y el 3 en la posición 0 de la tabla 2.

table[1]	-	67	-	36	-	-	105	-	-	53	-
table[2]	3	20	-	-	50	-	75	-	-	100	-

Dispersión del cuco

Valores de las funciones de dispersión:

h1(key) = key%11

h2(key) = (key/11)%11

	20	50	53	75	100	67	105	3	36	39
h_1(key)	9	6	9	9	1	1	6	3	3	6
h_2(key)	1	4	4	6	9	6	9	0	3	3

Finalmente se inserta el 39. Quedando el 39 en la tabla 1, el 105 en la tabla 2, el 100 en la tabla 1, el 67 en la tabla 2, el 75 en la tabla 1, el 53 en la tabla 2, el 50 en la tabla 1 y el 39 en la tabla 2.

table[1]	-	100	-	36	-	-	50	-	-	75	-
table[2]	3	20	-	39	53	-	67	-	-	105	-